
Caravel : installation et configuration

Document annexe

Mendes Reis Steve

25 juillet 2021

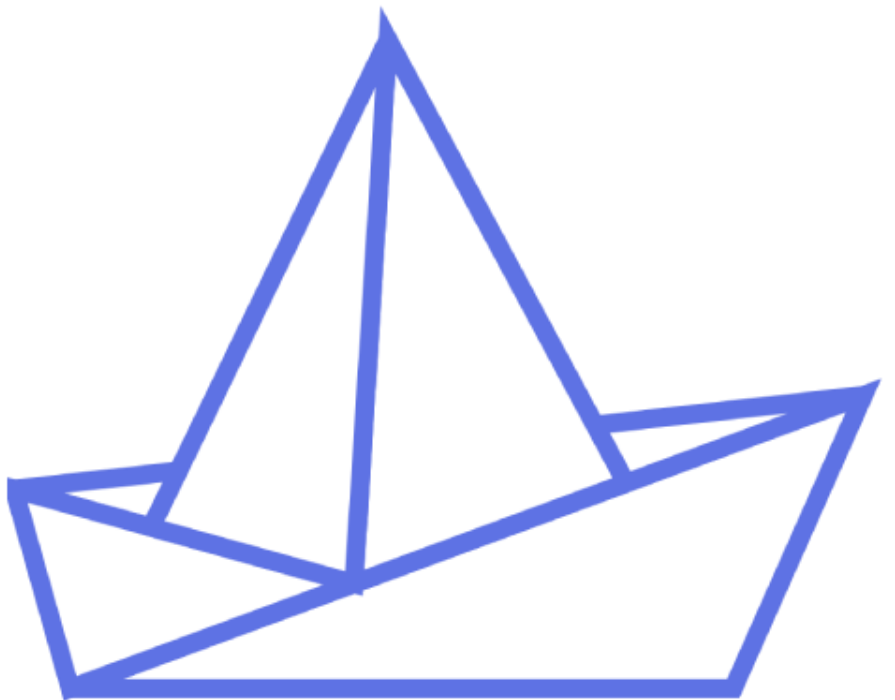


Table des matières

| | | |
|----------|--|----------|
| 1 | Introduction | 5 |
| 2 | Pré-requis | 5 |
| 3 | Installation | 5 |
| 3.1 | Backend | 5 |
| 3.1.1 | Configuration de l'environnement | 5 |
| 3.1.2 | Installation | 6 |
| 3.2 | Frontend | 7 |
| 3.2.1 | Configuration | 7 |
| 3.2.2 | installation | 7 |
| 4 | Lancement normal | 7 |

Table des figures

Liste des codes

| | | |
|---|--|---|
| 1 | configuration de la connexion LDAP | 6 |
|---|--|---|

1 Introduction

Ce document explique en détail l'installation et la configuration du projet Caravel.

2 Pré-requis

Les sections suivantes ont besoin que les éléments qui suivent soient installés sur le système cible :

- Composer¹
- NPM²
- MariaDB ou PostgreSQL
- Git
- PHP (version >= 7.4)
- Accès LDAP

3 Installation

Dans un premier temps il faut cloner le projet depuis le GitHub avec la commande

```
git clone https://github.com/HE-Arc/Caravel.git .
```

Une fois le projet cloné, deux dossiers vont particulièrement nous intéresser, ce sont les dossiers frontend et backend.

3.1 Backend

Naviguez dans le dossier backend

```
cd backend
```

3.1.1 Configuration de l'environnement

Avant de lancer toutes les commandes nécessaire, il faut configurer l'application, pour cela éditez le fichier `.env` et modifiez les éléments selon votre configuration, notamment les variables de DB, LDAP ainsi que les informations sur l'application tel que `APP_URL`.

¹<https://getcomposer.org/>

²<https://www.npmjs.com/>

3.1.1.1 LDAP Pour la partie LDAP, si vous ne disposez pas d'un LDAP, vous pouvez laisser la configuration par défaut et utilisez un LDAP de test

```
docker run -d --rm -p 10389:10389 -p 10636:10636 rroemhild/test-openldap
```

Lorsque le LDAP est configuré il est important de savoir de quelle implémentation il s'agit (Active Directory, OpenLDAP, etc...) car les connecteurs ne sont pas forcément les mêmes pour l'authentification.

Il faut donc éditer le fichier `config/auth.php` et adapter la ligne model

Code 1 : configuration de la connexion LDAP

```
1 ...
2
3 'providers' => [
4     'users' => [
5         'driver' => 'ldap',
6         'model' => LdapRecord\Models\OpenLDAP\User::class, // ligne à
                        adapter
7         'rules' => [],
8
9     ...

```

3.1.2 Installation

Dans un premier temps il faut installer toutes les dépendances nécessaires au bon fonctionnement du backend avec composer

```
composer install
```

Une fois l'installation terminée, il faut créer la base de donnée avec la commande

```
php artisan migrate
```

La base de donnée a besoin d'être nourrie pour les types de tâches, il faut donc utiliser un seeder avec la commande suivante :

```
php artisan db:seed --class=TaskTypeSeeder
```

L'installation est terminée

pour vérifier que tout fonctionne, on peut lancer directement le serveur avec la commande

```
php artisan serve
```

3.2 Frontend

Il faut se rendre maintenant dans le dossier du frontend, si vous étiez dans le dossier backend faites un :

```
cd ../frontend
```

3.2.1 Configuration

Comme pour le backend il existe à la racine du projet un fichier `.env` qu'il s'agit d'éditer pour faire correspondre la configuration à votre contexte.

3.2.2 installation

Les commandes sont assez simple pour la partie frontend, il suffit de lancer l'installation avec la commande

```
npm install
```

puis lorsque l'installation est terminée on peut lancer le serveur avec la commande suivante :

```
npm run serve
```

4 Lancement normal

Une fois l'installation terminée le lancement de l'application se résume à 2-3 commandes :

```
cd backend  
php artisan server
```

pour le backend

```
cd frontend  
npm run serve
```

Eventuellement si vous avez besoin d'un serveur LDAP il faut lancer

```
docker run -d --rm -p 10389:10389 -p 10636:10636 rroemhild/test-openldap
```